Finding Differentially Expressed Genes Through Noise Elimination

Anne Denton and Angshu Kar
Department of Computer Science
North Dakota State University
Fargo, ND 58105-5164

Abstract

An algorithm for identifying differentially expressed genes in data from time course experiments is introduced. We evaluate our algorithm on a real microarray data set, in which 6% of genes have been identified as non-random based on expert knowledge. Our algorithm is related to subspace clustering based on axis-parallel projections. In contrast to existing subspace clustering techniques we sum over all possible 2^d combinations of dimensions rather than identifying the locally most relevant dimensions. We scale attribute values based on rank order to avoid making assumptions on the distribution of random data. Information on the overall expression of a gene contributes towards the final result. We demonstrate that our algorithm consistently outperforms a conventional outlier detection algorithm.

Keywords: Outlier Analysis, Noise, Gene-Expression Analysis, Density-based Clustering

1 Introduction

Two types of gene expression analysis are typically distinguished. Statistical techniques are used to test if a gene is involved in a particular process [16]. Experiments are replicated under identical conditions, and the significance of the observed expression is calculated. Time course experiments provide information on how the expression of genes changes over time. Results are typically analyzed using clustering techniques [18, 17, 5, 27, 31]. In practice, that means that biologists have to budget for two sets of experiments. First they have to use replication to develop the subset of genes that are relevant to their process of interest. Second, they have to do time course experiments to then cluster the subset of genes.

Techniques that can identify differentially expressed genes from time course data could allow biologists to use more of their resources for time course experiments and benefit from the richer information they provide. However, standard clustering techniques are not up to this challenge. In fact, some algorithms that are popular in the clustering of gene expression data, in particular k-means [23], cannot distinguish noise at all, and cannot be used for the purpose. Density-based clustering techniques [19, 21, 13] do have the capability of handling outliers and can, therefore, in principle discard genes that should not be part of a cluster. However, they are not optimized for doing so. Rather it is considered the domain of outlier detection algorithms to distinguish data that belong into clusters, from those that don't.

In this paper we present an outlier detection algorithm that uses subspace clustering concepts and apply it to a gene expression data set from cell cycle experiments by Cho et al. [12] (Cho data set). We compare our algorithm with a conventional density-based outlier detection algorithm that is loosely related to [28]. We use two lines of reasoning. Following the clustering logic, one would expect that differentially expressed genes fall into clusters, while others do not. In this type of reasoning those genes should be outliers that are not differentially expressed. Such a reasoning can be used for finding differentially expressed genes if a normalization is chosen that places all time course results on a hypersphere and does not consider the magnitude of expression, which is typically done in the clustering of gene expression data.

One could also argue that differentially expressed genes should be considered outliers themselves, since their expression values are expected to be larger than those of genes that are not differentially expressed. The normalization process mentioned above discards information related to the overall magnitude of expression. We, therefore, do a second comparison, in which we use density-based outlier detection without normalizing on a per-gene basis. The results we get in this fashion are substantially better than the ones based on the reasoning of high cluster density. However, we show that the performance is comparable to simply using the overall norm of the expression vector as indicator. When combining the ranking of genes based on norm with the ranking based on our subspace algorithm we get the overall best result.

One of the main challenges in this work is that outlier detection algorithms are normally intended for settings, in which the outliers are exceptions. In gene expression experiments it is, however, common that only a small set of genes is involved in the process of interest. In the Cho data set only about 6% of genes are involved in the cell-cycle. If interesting genes are to be discovered through high density in attribute space then it is very important to find a normalization that results in a constant landscape for noise. In this paper we assume that we neither have a noise model [15] nor labeled samples of noise [25]. When little is known about a distribution, a common approach in statistics is to use rank order of attributes instead of numerical values. Following this concept, we scale attributes such that the distribution of each attribute becomes constant. Density fluctuations can still occur in the space spanned by multiple attribute. Those fluctuations will determine whether we consider a point random.

A second challenge is that high density may only be associated with some of the dimensions. This problem is discussed extensively in the subspace clustering literature [29, 4, 1]. Subspaces can either be axis-parallel [4] or general projections of the high dimensional space [1]. Comparing densities for subspaces of varying dimensionality is, however, not straightforward. We avoid this problem by not selecting subspaces, but rather summing over all possible axis-parallel projections of a hypercube of a given side length. If a point is close in many dimensions, that point will also be close in many projections.

It is important to note that it would not be practical to sum over all possible subspaces, since the number of axis-parallel subspaces of a d-dimensional space grows as 2^d. Hence, it is critical for the performance of the algorithm to circumvent computations that involve individual subspaces. The complexity of our algorithm and of an equivalent algorithm on the full space only differ by the need for sorting arrays of length d. This operation can be expected to have complexity $d \log d$. Our algorithm returns a ranking of data from most likely non-random to least likely and we use ROC evaluation [34] to measure success. While this setup resembles classification with class labels *non-random* and *random*, it is important to note that no training is involved. We do not need random samples to define randomness [25]. Our concept of randomness is entirely based on the notion of attributes being independent.

2 Related Work

Outlier detection is an important problem in statistics and data mining. The term outlier is used to describe individual attribute values that are exceptional [26, 30] or complete data points [24, 2, 25, 36]. Outlier detection or noise removal may be done as part of data cleaning [30, 36] or outliers may be of interest by themselves, such as in fraud detection [24, 2].

In gene expression data analysis, statistical techniques based on a single set of experimental conditions are conventionally used for determining relevance of a gene to a process [16]. Some clustering algorithms detect noise as a byproduct [18]. Biclustering techniques [11], which cluster genes and experiments simultaneously, often also have the capability of excluding noise. Special techniques have been developed for gene expression data that treat similar genes as exceptional case and are therefore related to our approach [8, 20]. These techniques make use of specific properties of time course gene expression experiments and are hence not as general as the approach discussed in this paper.

Gene expression data from yeast cell cycle experiments [12, 33] have been used in many clustering papers [6, 22, 32]. Most of those apply clustering algorithms only to genes, which have identified as cell cycle genes [38, 6, 32]. Kailing et al. [22] apply subspace clustering to the complete set of genes and recognize several cell-cycle genes in clusters. However they do not quantitatively test the selectivity of their algorithm.

The normalization used in this paper is related to quantile normalization [9] in that the quantile of a point determines its normalized value. In our normalization we use a constant distribution as reference distribution. Such a normalization results in distances that are closely related to the mass-distance discussed in [39]. We go beyond pairwise comparisons and evaluate the collective similarity of a given gene to all other genes within a neighborhood that is determined by the algorithm. The latter step is similar to density-based clustering [21, 13, 19] and subspace clustering [29, 7, 1, 10, 3] with a rigorous comparison against a noise model [15, 14]. It differs from the cited approaches in its collective evaluation of all possible subspaces.

3 Concepts

3.1 Rank-order Scaling The more we know about random data, the better we can distinguish non-random data. Gene expression data do not typically follow a Gaussian distribution. Figure 1 shows the distribution of individual attribute values of genes that are not expressed in the Cho data set together with a normal distribution of equal variance. It can be seen that the real data set has a narrower peak with much longer tails. Cell-cycle genes are not included in this plot to highlight the non-Gaussian behavior of random data.

In this paper we normalize each attribute using rank order. That means, that by definition the density in each dimension is constant. Fluctuation can happen when attributes are correlated. Strictly, we would prefer continuing to use only data that correspond to noise for our scaling process. However, the information on which data are noise is assumed not to be known. Hence the scaling has to be done based on the full data. Since we are primarily interested in the case of massive noise we can assume that noise dominates the scaling. For given attribute values, rank order provides a simple measure of size that satisfies the requirement of following a constant distribution. For convenience we work with data in the range $-0.5 < x_i < 0.5$. Using the Heaviside step function θ we can formally write the scaled attribute value x'_i as a function of the unscaled values x_j as

$$x'_i = \frac{\sum_{j=1}^{N} \theta(x_i - x_j)}{N} - 0.5 \tag{3.1}$$

where N is the total number of values of the attribute.

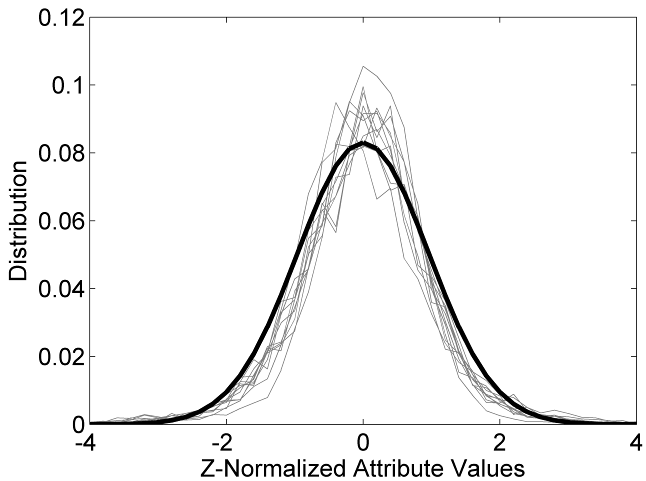

Figure 1: Distribution of attributes in gene expression data set (only genes that are not differentially expresse), compared with normal distribution.

3.2 Confidence Optimization We are interested in data points for which the density most differs from what we would expect for random data. For this purpose, we define a confidence measure as follows

$$c(a) = \frac{\frac{\rho_{\text{obs}}(a)}{\rho_{\text{exp}}(a)}}{1 + \frac{\rho_{\text{obs}}(a)}{\rho_{\text{exp}}(a)}} = \frac{\rho_{\text{obs}}}{\rho_{\text{obs}} + \rho_{\text{exp}}} \quad (3.2)$$

where $\rho_{\text{obs}}(a)$ is the observed number of points per volume, and $\rho_{\text{exp}}(a)$ is the number that is expected based on a random distribution. $c(a)$ can be seen as our confidence that a data point within the volume belongs to the data rather than being noise. It is closedly related to the multi-granularity deviation factor, MDEF in [28], but increases for increasing density where MDEF decreases. In practice, a grid of $a/2$ values is considered, also similar to [28], typically starting at $a/2 = 0.025$, and continuing at equidistant intervals through $a/2 = 1$. The confidence of equation (3.2) is calculated for each value of $a/2$. For each data point $\mathbf{x}$ the highest confidence is selected.

Figure 2 illustrates the volume calculation for projections of a two-dimensional "hypercube", i.e. square. Note that we do not allow the side length of the evaluation volume to differ in different dimensions. Allowing more generality would also increase the probability of finding a hypervolume that appears as relevant because of random fluctuations alone. The analytical representation of the sum of the volumes in all subspaces can be derived by observing that each dimension may either be involved in defining the volume, or it may not be involved. This choice exists independently for each dimension. Hence, the contribution of the dimension being involved or not involved can be added independently for each dimension. The following equation gives the sum

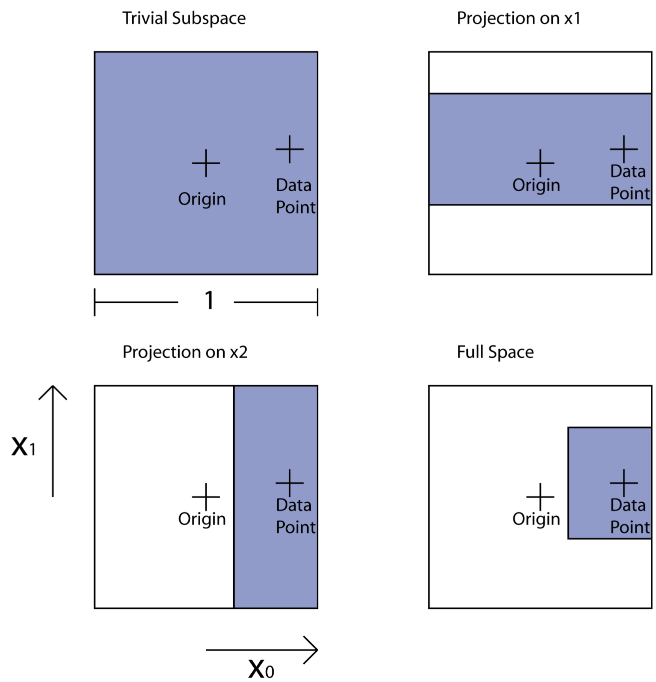

Figure 2: Subspace setup in two dimensions x_0 and x_1. Top left shows the trivial subspace, that results by performing projections for all dimensions. Right bottom represents the original space.

of volumes over all dimensions

$$\rho^{(s)}_{\text{exp}}(\mathbf{x}, a) \sim \prod_{i=1}^{d} \left[1 + \min\left(x_i + \frac{a}{2}, \frac{1}{2}\right) - \max\left(x_i - \frac{a}{2}, -\frac{1}{2}\right) \right] \quad (3.3)$$

Note that the only difference between the sum over all subspaces and the volume of the full space alone is the constant contribution of 1 to each term. The difference looks trivial, but it means that equation (3.3) simultaneously considers 2^d subspaces instead of only the complete space. We can immediately observe that the calculation of the expected number of points in all projections is no more computationally expensive than the computation of the expected number of points in the one hypercube defined for the full space.

The calculation of the actual number of data points within all projections can also be done without enumerating subspaces. Note that a data point only has to be within distance $a/2$ for dimensions, over which no projection was performed. Let us look at some examples:

- If the data point is closer than $a/2$ for all its dimensions, then it is within the hypercube for each of the 2^d subspaces.

Table 1: Subspaces for d=3

$\lvert x_1 - x_1^0\rvert < \lvert x_2 - x_2^0\rvert < \lvert x_3 - x_3^0\rvert$			
0	0	0	trivial subspace contributes for all data points
1	0	0	contributes for $\lvert x_1 - x_1^0\rvert < a/2$
0 1	1 1	0 0	contribute for $\lvert x_2 - x_2^0\rvert < a/2$
0 1 0 1	0 0 1 1	1 1 1 1	only contribute for $\lvert x_3 - x_3^0\rvert < a/2$

- If there is one dimension for which the distance is greater than $a/2$, then the point is only in 2^{d-1} subspaces.
- If the data point only has a single dimension d_i for which it is closer than $a/2$ then the data point is only in two subspaces, the trivial subspace that results by performing projections for all dimensions, see top left in Figure 2, and the subspace is projected on d_i.
- Every data point is part of the trivial, fully projected subspace, in which the hypercube does not provide any constraint, and that has the constant volume of 1.

Table 1 shows the subspaces for $d = 3$, and lists which of them contribute for which values of the point differences. Note that there are only $d + 1$ possible weights for any data point. With $|x_1 - x_1^0| < |x_2 - x_2^0| < |x_3 - x_3^0|$ the following weights can occur

$a/2 < \lvert x_1 - x_1^0\rvert$	1
$\lvert x_1 - x_1^0\rvert < a/2 < \lvert x_2 - x_2^0\rvert$	2
$\lvert x_2 - x_2^0\rvert < a/2 < \lvert x_3 - x_3^0\rvert$	4
$\lvert x_3 - x_3^0\rvert < a/2$	8

An expression for the number of subspaces, for which a data point at $\mathbf{x}$ is within range, given a hypercube of side length a centered on a data point $\mathbf{x^0}$ is hence

$$w = \max_{\substack{i=0 \\ |x_i - x_i^0| < a/2}}^{d} 2^i \tag{3.4}$$

The resulting algorithm for evaluating $\rho_{\mathrm{obs}}^{(s)}(\mathbf{x}, a)$ for a grid of values $a_j/2$ summed over all subspaces, is as follows:

For all data points, and for all grid-values $a_j/2$

1. Calculate the vector of differences $\mathbf{x} - \mathbf{x}^0$ with respect to the central point $\mathbf{x}^0$
2. Take the absolute value of the differences separately for each dimension $|x_i - x_i^0|$
3. Sort difference values within each data point, from smallest to largest
4. Add 2^i to the count of subspaces for the largest i for which the difference vector is smaller than the grid value $a_j/2$
5. Return count of subspaces as function of $a_j/2$

It is important, in this algorithm, that the central data point is not itself included in the sum over subspaces. Since the central point has a distance of 0 in every dimension, its weight is 2^d, which would distort the result substantially, since ρ_{exp} has no corresponding contribution.

4 Evaluation

4.1 Methods The algorithm was implemented in MATLAB. For the evaluation on microarray data we used results of cell cycle experiments in yeast described in [12] and available as part of the data distributed with [33] (cdc28 measurements). We used the subset of genes from [37] that have been identified as relevant by expert inspection. The genes are discussed as having recognizable features that make them similar to each other and relevant to the cell cycle experiments that were performed on the yeast genome. The data set has a total of 6178 genes, of which 383 are involved in the process tested in the experiment and the remainder are considered noise. This corresponds to about 94% noise. We also evaluated the algorithm for other fractions of noise, each time using the full set of relevant genes. There were 17 attributes within the cdc28 experiments. Some of them, however, had a substantial number of missing data. We filled in missing values with zeros, before rank order normalization. This choice worked well as long as there were fewer than about 5% values of unkown values within an attribute. We excluded attributes that had more missing values, which left us with the first 11 dimensions of the data set. We evaluated hypercubes with side lengths from 0.05 through 2, since a hypercube centered on the perimeter of the data hypercube of side length 1 must have side length 2 to cover all points.

Our algorithm returns a ranking of genes from highest to lowest density. We then use ROC analysis to evaluate the effectiveness regardless of cut-off. Seven data sets were created with different fractions of noise. Each data set consists of all cell-cycle genes and a varying number of non-cell-cycle genes. The noise fraction is defined as the number of noise (non-cell-cycle) genes divided by the total number of genes considered. It should be stressed that while the evaluation is done in analogy to classification, the problem as such was not set up as a classification problem. In particular, there is no training based on expert opinion. The algorithm does not require input of any explicit

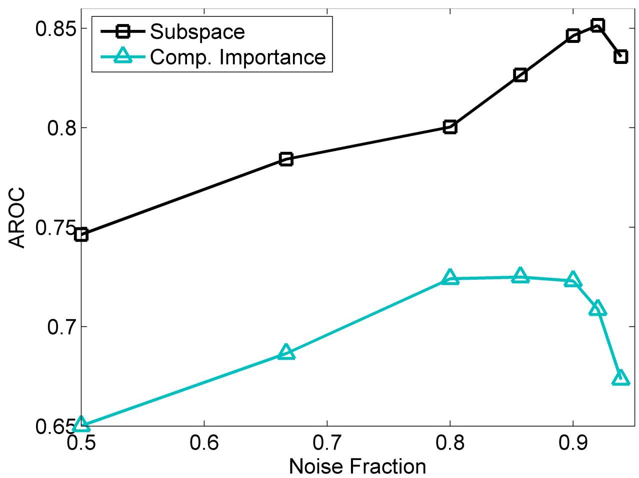

Figure 3: Area under the ROC curve for different noise fractions up to the full data set (94% noise)

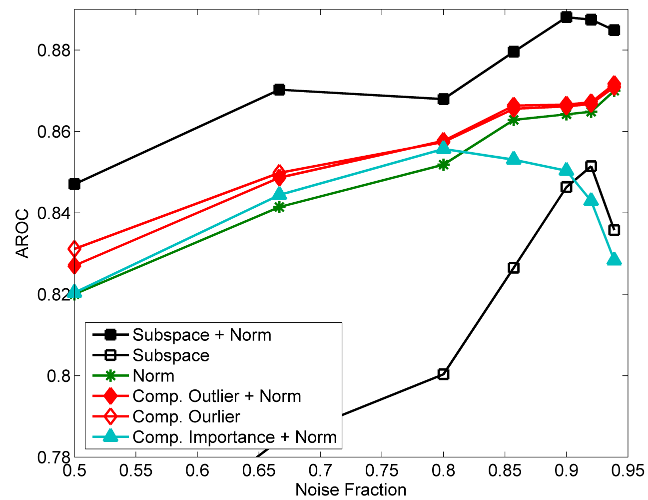

Figure 4: Area under the ROC curve for individual algorithms and for averaged rankings of the norm of expression profiles (”Norm”) the output of the algorithms. For ”Comp. Importance” by itself please refer to Figure 3.

parameter values. However, a heuristic has to be determined for the minimum number of points that qualify within the confidence optimization of equation (3.2). We use the heuristic that every point has to contribute on average one subspace to the sum.

The comparison code uses ideas from [28] but does not implement the full correlation integral for the following reasons: We are working with a relatively high-dimensional space of 11 dimensions, where the difference in distance between the closest and the furthest neighbors is small [35], and a distinction between a counting and a sampling neighborhood is not practical. Rather, we sample over the entire system and then evaluate hyperspheres for 20 different radii and pick the maximum of the actual number of points divided by the average number of points. Neither do we need the statistical determination of a cutoff, since we use ROC-based evaluation that does not require setting a cutoff.

4.2 Comparison with Conventional Density-Based Outlier Detection In our first comparison, we will maintain the assumption that differentially expressed genes should be recognized by being in a dense region of similarly expressed genes. This perspective is an extension of standard clustering of gene expression data, in which we consider not differentially expressed genes as outliers. We normalize the expression profile of each gene by subtracting the mean and dividing by the standard deviation. Figure 3 shows the area under the ROC curve for our subspace algorithm together with the ranking based on high actual density compared with its expected value. It can be seen that the subspace algorithm outperforms the comparison code for all noise fractions from 50% (equal number of genes that are differentially expressed and genes that are not) through 94% (full data set).

In our second comparison we assume that differentially expressed genes might themselves be considered outliers since they have exceptionally large expression values. In this comparison we only apply an attribute-based Z-normalization. For this purpose we use a ranking that places points with sparse neighborhoods highest. In Figure 4 this is approach is labeled as ”Comp. Outlier”. It can be seen that this approach leeds to substantially better results than the pure subspace algorithm. It is not surprising that large expression values signify differentially expressed genes. We, therefore, also compare the predictions with a ranking based on the overall norm of expression profiles. It can be seen that the resulting ”Norm” ranking in Figure 4 is an almost as good predictor as ”Comp. Outlier”.

We then combined the ”Norm” ranking with other rankings by averaging. It can be seen that the average of the ”Subspace” and the ”Norm” ranking produces the clearly best-performing result. It can also be seen that the ”Comp. Importance” result is substantially improved over Figure 3 by averaging with the ”Norm” ranking. However, averaging the ”Comp. Outlier” ranking with the ”Norm” ranking produces a result that barely differs from the ”Comp. Outlier” alone. This observation confirms that both capture very similar information. Averaging of classifiers is only expected to improve classification when the classifiers capture different information, and the same can be expected to hold in this analysis. Figure 5 shows the ROC curves for the full data set (noise fraction 94%). It confirms that the ”Subspace + Norm” ranking does indeed have True Positive results high up in the ranking as is expected of an overall useful algorithm.

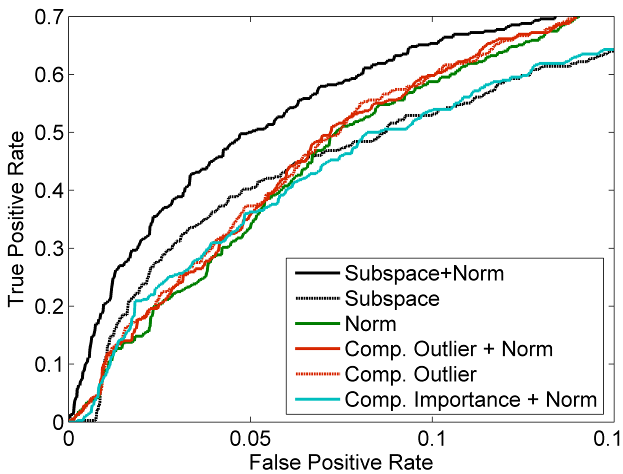

Figure 5: ROC curves for the same rankings as in Figure 4.

The scaling of our algorithm with data set size is governed by the need for examining each data point, and for each data point examining the distribution of neighbors, and is thereby quadratic in the number of data points. For each pair of data points we also have to sort the distances in all dimensions, contributing a factor of $d \log d$. In this work we primarily focused on effectiveness and therefore did not explore opportunities for improving performance.

5 Conclusions

We have presented an algorithm for finding differentially expressed genes from time course experiments in the presence of up to 95% noise, i.e. not differentially expressed genes. Our rank-order-based scaling creates a flat distribution for each individual attribute, regardless of the distribution of values. Differentially expressed genes are identified through density maxima that result from the combined occurrence of similar attribute values in multiple data points. Similarity is evaluated over all axis parallel subspaces. The side length of the evaluation hypercube is optimized based on the confidence that a data point is non-random. The overall expression is considered by averaging over the ranking according to similarity and a norm. We demonstrate the effectiveness of the resulting algorithm on a real gene expression data set, for which noise has a clearly non-Gaussian distribution. We show that the algorithm outperforms a conventional outlier detection algorithm for noise fractions from 50% to 94% .

Acknowledgements

Many thanks to Dr. Daniel Keim for very valuable comments. Thanks also to Dietmar Dorr and Christopher Besemann for their contributions to discussions. This material is based upon work supported by the National Science Foundation under Grant No. 0415190.

References

[1] C. C. Aggarwal and P. S. Yu. Finding generalized projected clusters in high dimensional spaces. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 70–81, 2000.

[2] C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. In *Proc. SIGMOD Conference*, 2001.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data*, pages 94–105, 1998.

[4] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data. *Data Mining and Knowledge Discovery Journal*, 11(1), 2005.

[5] Z. Bar-Joseph. Analyzing time series gene expression data. *Bioinformatics*, 20(16), 2004.

[6] Z. Bar-Joseph, G. Gerber, T.S. Jaakkola, D.K. Gifford, and I. Simon. Continuous representations of time series gene expression data. *J Comput Biol.*, 10(3-4), 2003.

[7] C. Baumgartner, K. Kailing, H.-P. Kriegel, P. Kröger, and C. Plant. Subspace selection for clustering high-dimensional data. In *Proc. 4th IEEE Int. Conf. on Data Mining (ICDM'04)*, pages 11–18, Brighton, UK, 2004.

[8] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering local structure in gene expression data: the order-preserving submatrix problem. In *RECOMB '02:Proc 6th annual international conference on Computational biology*, New York,NY, 2002.

[9] B. M. Bolstad, R. A. Irizarry, M. Astrand, and T.P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on bias and variance. *Bioinformatics*, 19(2):185–193, 2003.

[10] C.H. Cheng, A. Wai-Chee Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *Knowledge Discovery and Data Mining*, pages 84–93, 1999.

[11] Y. Cheng and G.M. Church. Biclustering of expression data. In *Proc. 8th Int'l Conf. on Intelligent Systems for Molecular Biology (ISMB)*, pages 93–103, 2000.

[12] R.J. Cho, M.J. Campbell, et al. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2(1), 1998.

[13] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[14] A. Denton. Density-based clustering of time series subsequences. In *Proc. 3rd Workshop on Mining Temporal and Sequential Data (TDM 04) in conjuction with ACM SIGKDD*, Seattle, WA, 2004.

[15] A. Denton. Kernel-density-based clustering of time series subsequences using a continuous random-walk noise model. In *Proc. 5th IEEE Int. Conf. on Data Mining (ICDM'05),*, pages 122–129, Houston, TX, 2005.

[16] S. Dudoit, Y.H. Yang, T.P. Speed, and M.J. Callow. Statistical methods for identifying differentially expressed genes in replicated cdna microarray experiments. *Statistica Sinica*, 12(1):111–139, 2002.

[17] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein.

Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci.*, 95, 1998.

[18] J. Ernst, G.J. Nau, and Z. Bar-Joesph. Clustering short time series gene expression data. *Bioinformatics*, 21(Supplement 1), 2005.

[19] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery*, 2(2):169–194, June 1998.

[20] B.J. Gao, O.L. Griffith, M. Ester, and S.J.M. Jones. Discovering significant opsm subspace clusters in massive gene expression data. In *2006 ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, Philadelphia,PA, 2006.

[21] A. Hinneburg and D.A. Keim. A general approach to clustering in large databases with noise. *Knowl. Inf. Syst.*, 5(4):387–415, 2003.

[22] K. Kailing, H.P. Kriegel, P. Kroeger, and S. Wanka. Ranking interesting subspaces for clustering high dimensional data. In *Proceedings of the PKDD Conference*, pages 241–252, 2003.

[23] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, New York, 1990.

[24] E.M. Knorr and R.T. Ng. A unified notion of outliers: Properties and computation. In *Knowledge Discovery and Data Mining*, pages 219–222, 1997.

[25] X. Liu, G. Cheng, and J.X. Wu. Analyzing outliers cautiously. *IEEE Transactions on Knowledge and Data Engineering*, 14(2):432–437, 2002.

[26] W. Mendenhall, J.E. Reinmuth, and R.J. Beaver. *Statistics for Management and Economics*. Duxbury Press, Belmont, CA, 1993.

[27] C.S. Moller-Levet, K.H. Cho, and O. Wolkenhauer. Microarray data clustering based on temporal variation: Fcv with tsd preclustering. *Appl. Bioinformatics*, 2(1), 2003.

[28] S. Papadimitriou, H. Kitagawa, P. Gibbons, and C. Faloutsos. Loci: Fast outlier detection using the local correlation integral. In *In Proc. 19th Int'l Conf. on Data Engineering (ICDE)*, pages 315–326, 2003.

[29] L. Parsons, H. Ehtesham, and H. Liu. Subspace clustering for high dimensional data: a review. In *ACM SIGKDD Explorations Newsletter*, volume 6, pages 90 – 105, 1998.

[30] D. Pyle. *Data Preprocessing for Data Mining*. Morgan Kaufmann, San Francisco,CA, 1999.

[31] M.F. Ramoni, P. Sebastiani, and I.S. Kohane. Cluster analysis of gene expression dynamics. *Proc. Natl Acad. Sci.*, 99(14), 2002.

[32] V. Roth, M. Braun, T. Lange, and J. Buhmann. A resampling approach to cluster validation. In *Computational Statistics (COMPSTAT'02)*, 2002.

[33] P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998.

[34] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2006.

[35] M. Verleysen and D. François. The curse of dimensionality in data mining and time series prediction. In J. Cabestany, A. Prieto, and F. Sandoval, editors, *Computational Intelligence and Bioinspired Systems, Lecture Notes in Computer Science 3512*, pages 758–770. Springer, 2005.

[36] H. Xiong, G. Pandey, M. Steinbach, and V. Kumar. Enhancing data analysis with noise removal. *IEEE Transactions on Knowledge and Data Engineering*, 18(3), 2006.

[37] K.Y. Yeung. *Cluster Analysis of Gene Expression Data*. Dissertation, University of Washington, 2001.

[38] K.Y. Yeung, D. R. Haynor, and W. L. Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 17, 2001.

[39] G. Yona, W. Dirks, S. Rahman, and D.M. Lin. Effective similarity measures for expression profiles. *Bioinformatics*, 22(13):1616–1622, 2006.